Eur päisches Patentamt

European Patent Office

Office européen des brevets

Bescheinigung

Certificate

Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. **Patent application No.** **Demande de brevet n°**

02368100.0

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

DEN HAAG,DEN
THE HAGUE, 09/10/02
LA HAYE,LE

Europäisches Patentamt | European Patent Office | Office européen des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.: 02368100.0
Demande n°:

Anmeldetag:
Date of filing: 12/09/02
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):
INTERNATIONAL BUSINESS MACHINES CORPORATION
Armonk, NY 10504
UNITED STATES OF AMERICA

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
Method and systems for encoding signatures to authenticate files

In Anspruch genommene Prioriät(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat: State: Pays:	Tag: Date: Date:	Aktenzeichen: File no. Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:
/

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE/TR
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

METHOD AND SYSTEMS FOR ENCODING SIGNATURES TO AUTHENTICATE FILES

Field of the Invention

The present invention relates to network computing security and more specifically to a method and systems for verifying the authenticity and integrity of files accessed and retrieved through a network.

Background of the Invention

Nowadays, a serious risk associated to the exchange of electronic information on open and unsecured networks, particularly on the Internet, concerns the modification of data during the transfer. As a consequence, it is important to authenticate files received over a network to verify that they have not been corrupted nor altered and/or that they have not been sent by an impostor. For example, when a user received a file attached to an e-mail, such an authentication must be done when clicking on the file attachment icon. Such attached files may include, computer programs, text documents, graphics, pictures, audio, video, or other information that is suitable for use within a computer system. Likewise, if a document includes a link to an executable file or a software program, the user may wish to ensure that the received file has been sent by a trustworthy party prior to exposing his computer system to a program file that might include a "Trojan Horse" or that could infect the user's computer with a virus. As a result, the demand of secured transfer system increases.

To improve data transmission security over computer networks and prevent digital forgery, digital signature is commonly used to provide document and signer authentication, i.e. to control the source of a received file, and verify document integrity. Digital signatures are based upon cryptographic-key based algorithms wherein security is provided through one or several keys independently of the algorithm that may be freely published or analyzed. Two general types of key based authentication algorithms are well known in the art to authenticate any kind of digital document : symmetric and public-key.

In symmetric algorithms the encryption key and the decryption key are the same and must be kept in secrecy by both parties, the sender and the receiver. The standard solution consists in adding a Message Authentication Code (MAC) to the transmitted documents. The MAC is computed with a one-way hash function over the document and depends on the secret key known by the sender and the receiver. The MAC allows the receiver to check that what he has received has been sent by whom shares the secret-key with him and the document has not been altered. For example, the Secure Hash Algorithm (SHA) specified by the National Institute of Standards and Technologies (NIST), FIPS PUB 180-1, "Secure Hash Standard", US Dpt of Commerce, May 93, produces a 160-bit hash value. It may be combined with a key, e.g. through the use of a mechanism referred to as Keyed-Hashing for Message Authentication (HMAC), subject of the Request For Comment (RFC) of the Internet Engineering Task Force (IETF) under the number 2104. HMAC is devised so that it can be used with any iterative cryptographic hash function including SHA. Therefore, a MAC can be appended to the transmitted document so that the whole document can be checked by the receiver.

Public key algorithms, also known as asymmetric algorithms, are using two different keys, one is used for signing and the second one is used for verification. These algorithms are called "public-key" algorithms because the verification key can be made public. In contrast, the signature key needs to be kept secret by its owner, the signer.

Using digital signatures involves two processes, one performed by the signer to generate the signature and the other by the receiver that consists in verifying the signature. The signer creates a digital signature for a particular document by using his private key and transmits both the document and the digital signature to the receiver. Verification process consists in checking the digital signature received with the document using the public verification key. Properties of cryptographic digital signatures are such that there is no way of extracting someone's digital signature from a document and attach it to another. Likewise, any change in the signed document are detected since any change will cause the signature verification process to fail. Furthermore, the signing key can not be calculated from the verification key in a reasonable time.

In practical implementations, public-key algorithms are generally not adapted to determine signature on long documents. Thus, to save time, signature protocols like Rivest-Shamir-Adleman algorithm (RSA) or Digital Signature Algorithm (DSA) are often implemented with secure (one-way) hash functions. Basically, instead of signing a complete document, the signer computes a hash value of the document and signs the computed hash value.

Several signature algorithms are in use today. One popular signature algorithm is a combination of a hashing algorithm and an RSA encryption algorithm, e.g.

Message-Digest-5 (MD5) with RSA and SHA with RSA. Another popular signature algorithm is the DSA encryption algorithm. The DSA encryption algorithm, which is available from the United States Government, may be used for limited purposes by private parties as a signature algorithm. Applied Cryptography, Second Edition, 1996, by Bruce Schneier which is available from John Wiley & Sons, Inc. New York City, N.Y., presents a detailed description of signature and hashing algorithms and related encryption operations.

Once the digital signature of a file has been computed, it must be associated with the signed file. Digital signatures authenticating a file can be appended to the file they authenticate, e.g. as part of a file wrapper structure, embedded within the file or transmitted as separate files. Each of these methods present advantages and drawbacks :

- wrapping a file with delimiters and appending the digital signature at the end of the file is convenient since both the signature and content travel together and algorithms to sign and check signatures are simple and efficient. Conversely, wrapper and signature will typically have to be removed before the file can be used. Thus, signature validation only occurs when the document is retrieved. If the document is later passed on or moved, it may be difficult to check it again. Furthermore, such method is not compatible with standard file formats such as image, video, audio or executable files that can not be recognized prior to authentication.

- embedding digital signatures into the files has received considerable attention to protect copyrights attached to digital multimedia materials that can be easily copied and distributed everywhere through the Internet and networks in general. A review of data embedding and data hiding techniques is described in "Techniques for data hiding" by W. Bender and al. IBM Systems Journal, Vol. 35, Nos 3&4, 1996. The most common form of high bit-rate encoding on images, reported in the mentioned publication, is the replacement of the least significant luminance bits of image data with the embedded data so that the alteration of the image is imperceptible. This method is used for watermarking or tamper-proofing to detect image alterations. However a first drawback lies in the lack of standardization about how and where integrating signatures into the different file formats, particularly on image, video, audio or executable files, and the added complexity of authenticating algorithms. Another important drawback is that merged checking information and file content affects readability and quality of documents, e.g. digital images.

- maintaining signatures and data in separate files, e.g. signature files may be stored on a server, presents the advantage of supporting file authentication at any time in a simple and well understood way. However, the signature can be intentionally removed in an attempt to cheat, lost or accidentally removed.

A more complex situation arises when authentication concerns a group of files, e.g. a document including attachments or links to other files. To deal with these frequent cases, a standard solution consists in aggregating the files and generating a single MAC by applying a cryptographic hashing algorithm on it. But such solution presents a main drawback since the receiver must authenticate all the files that are aggregated, which is time consuming. To remedy this problem, other methods provide, along with the group of files, a separate signature file or MAC file. This MAC file includes individual check-values for the files e.g. hash-values, as well as a digital signature or a MAC value for the group of files. Check-values of the signature file are compared with the corresponding values computed from the received files and the digital signature of the group of files is verified. A classical method for generating a separate signature file for groups of data files is described in US patent 5,958,051 "Implementing digital signatures for data streams and data archives", Renaud et al.. However, such methods using a separate signature file present several drawbacks as described above. Furthermore, if a file linked to the group have been withdrawn or is no more accessible, not any file of the group may be authenticated.

Therefore, there is a need for an efficient method and systems for securing and verifying the authenticity and integrity of all types of files so as to remedy the short-comings as discussed above.

Summary of the Invention

Thus, it is a broad object of the invention to remedy the shortcomings of the prior art as described here above.

It is another object of the invention to provide a method and systems to authenticate all types of files and groups of files without appending nor embedding digital signatures in the authenticated files.

It is a further object of the invention to provide a method and systems to authenticate all types of files and groups of files without encoding digital signatures on separate files.

The accomplishment of these and other related objects is achieved by a computer file containing digital data characterized in that authentication information is encoded in the filename of said computer file at a predetermined position or using delimiters,

by a method for encoding authentication information in the filename of a computer file containing digital data, said method comprising the steps of :

- computing a hash value of said computer file ;

- computing a digital signature of said computed hash value using a private key of the sender ; and,

- encoding said computed digital signature in the filename of said computer file at a predetermined position or using delimiters,

and by a method for authenticating a computer file having a filename comprising authentication information according to the previous method, comprising the steps of :

- extracting said authentication information from said filename of said computer file, at a predetermined position or using delimiters ;

- recovering the encoded hash value of the computer file by using the public-key of the sender and extracted authentication information ;

- computing the hash value of said computer file by means of a hash function, identical to the one used for authentication information encoding ;

- comparing said encoded and said computed hash values ; and,

- if said encoded and said computed hash values are identical, processing said computer file else, if said encoded and said computed hash values are different, rejecting said computer file.

Further advantages of the present invention will become apparent to the ones skilled in the art upon examination of the drawings and detailed description. It is intended that any additional advantages be incorporated herein.

Brief Description of the Drawings

Figure 1 illustrates the method of the invention for authenticating a file by encoding the signature in the filename of the file, generating a "signed filename".

Figure 2 describes the method of the invention for verifying the authenticity and integrity of a received file by using the digital signature extracted from the "signed filename".

Figure 3 illustrates an example of the prior art where the integrity information of a group of files, formed by an electronic document that includes a plurality of file attachments, is encoded on a separate signature file.

Figure 4 illustrates how the method of the invention encodes and verifies the signatures or MAC of a master file and attached or linked files, by using the digital signatures extracted from the corresponding "signed filenames".

Detailed Description of the Preferred Embodiment

Basis of the invention consists in encoding a certification or digital signature of a file into a portion of its filename. Since filenames can be freely formed by generic alphanumeric strings on all operating systems, independently of the file type, format and content, files authenticated in this way may be of any form, including document files, source program files, text files, executable files, audio files, image files or video files.

File systems

It exists different types of file systems available for different operating systems. Each file system type has its own format and set of characteristics such as filename length, maximum file size and so on. For example, on Linux operating system, the most commonly used file system type is the Second Extended File system, also known as ext2fs. It allows filenames up to 256 characters. On Windows (registered trademark of Microsoft) the VFAT (Windows 95) and FAT32 (Windows 98) file system maximum length of filenames is 255 characters. NTFS and UNIX file system maximum length of filenames is 256 characters.

These maximum available filenames lengths make possible to encode into a filename the corresponding signature or MAC. For example, the MAC of a file can be computed by means of a secret key using the HMAC (Keyed-Hashing for Message Authentication) method with the SHA (Secure Hash Algorithm) hashing method, producing a 160-bit keyed hash string. This MAC can be encoded in the filename of the authenticated file as a string of 40 hexadecimals. In the detailed description, the signature is appended to the filename to form a "signed filename".

Authentication encoding method

Figure 1 illustrates an embodiment of the invention for encoding authentication information of a file 100, named

FNAME.EXT 105. In this figure, the signature of the file 100 is computed by using a combination of a hashing algorithm 110 to obtain a hash value 115 and an RSA encryption algorithm 120 such as MD5 with RSA or SHA with RSA that uses the computed hash value and a private key 125. Then, by encoding the computed signature 130 in the filename 105 of the file 100, e.g. by appending it to the original filename before the file extension .EXT, a "signed filename" 135 is generated so that "signed filename" 135 contains the signature 130 of the authenticated file 100.

The authentication encoding method of this example comprises the steps of :

- computing a hash value FILE-HASH 115 of the file 100 by means of hash function 110 ;
- computing a digital signature 130 of the file hash value 115 using private-key 125 of the sender ;
- encoding the computed digital signature 130 in the filename 105 of the file 100 at a predefined position or using delimiters to create "signed filename" 135 ;
- transmitting the authenticated file 100 using "signed filename" 135.

Since the signature of the file is encoded in the filename, not on the file body, the original (non authenticated) file and the authenticated file are identical, having both exactly the same format and content.

In a preferred embodiment, the digital signature has a determined size and is added to the filename, just before the file extension.

Authentication verifying method

Figure 2 illustrates one embodiment of the invention for verifying the authenticity and integrity of a received file 200, that comprises authentication information according to the procedure described by reference to figure 1, i.e. by encoding the digital signature of the file 200 into the filename 205.

The verification method of this example comprises the steps of :

- extracting the encoded digital signature 210 from the "signed filename" 205 of received file 200 ;
- recovering the encoded hash value FILE-HASH* 220 of the received file 200 by using the public-key 215 of the sender, associated to corresponding private-key 125, and extracted signature 210 ;
- computing the hash value FILE-HASH 230 of received file 200 by means of hash function 225, identical to hash function 110 used by sender to compute digital signature 130 ;
- comparing the computed hash value FILE-HASH 230 with the decoded hash value FILE-HASH* 220 ; and,
- if the computed hash value FILE-HASH 230 and the decoded hash value FILE-HASH* 220 are identical, processing the received file 200 as an authentic file else, if the computed hash value FILE-HASH 230 and the decoded hash value FILE-HASH* 220 are different, rejecting the received file 200 as being fake or corrupted.

Authenticating groups of files

Figure 3 illustrates an example of the prior art where the integrity information of a group of files, formed in this case by a master electronic document referred to as

MASTER.DOC 300 that includes a plurality of files attachments and/or links, in particular DISCLOS7.LWP, FIG 1.PRZ and FIG J.PRZ, referred to as 305-1, 305-2 and 305-i, respectively, is encoded in a separate signature file 310 named SIGNATURE.TXT. This signature file includes individual check-values for all the attached or linked files, e.g. hash values, MAC2 and MAC3, as well as a digital signature or a MAC value for the group of files, e.g. hash value and MAC1.

Using the method of the invention, checking information is associated with each attachment or hyperlink object and not to the document that contains said attachment and/or hyperlinked objects. Turning now to figure 4, there is illustrated a master file 400 and its associated "signed filename" 405 comprising the signature or MAC 410 of this master file without taking attached or linked files into account. In this example, three files named DISCLOS7.LWP, FIG 1.PRZ and FIG J.PRZ, referred to as 415-1, 415-2 and 415-j, respectively, are attached or linked to master file 400. Each attached or linked file, generically referred to as 415, is associated to a "signed filename", generically referred to as 420, comprising the file name and the corresponding digital signature or MAC value, generically referred to as 425.

For sake of illustration, the hexadecimal string 410

"E1FF603A95E38C04DB751D44A82DC2402EA8BEF9"

is the MAC of master file 400 with "signed filename" 405

MASTER-E1FF603A95E38C04DB751D44A82DC2402EA8BEF9.DOC

while the hexadecimal string 425-1

"D05A7B402E3F855AC9003BE84CD7285DA4F7DE26"

corresponds to the MAC of attached file 415-1, having the "signed filename" 420-1

DISCLOS7-D05A7B402E3F855AC9003BE84CD7285DA4F7DE26.PRZ.

MAC values are computed, for example, by means of a secret key using the HMAC (Keyed-Hashing for Message Authentication) method with the SHA (Secure Hash Algorithm) hashing method, that produces a 160-bit (40 hex length) hash value. In this example illustrated on figure 4, digital signatures have a fixed size and are located just before the file extension so that they could be easily extracted.

Thus, the method for encoding authentication information comprises the steps of :

- for the master file :
  - computing a hash value of the file 400 by means of hash function ;
  - computing a digital signature 410 of the computed file hash value using the private-key of the sender ;
  - encoding the computed digital signature 410 in the filename of the file 400 at a predefined position or using delimiters to create "signed filename" 405 ;
- for each file attached or linked to the master file :
  - computing a hash value of the file 415 by means of hash function ;
  - computing a digital signature 425 of the computed file hash value using the private-key of the sender ;
  - encoding the computed digital signature 425 in the filename of the file 425 at a predefined position or using delimiters to create "signed filename" 420 ;
- transmitting the authenticated master file 400 using "signed filename" 405 and the attached files or links 415 and associated "signed filenames" 420.

Upon reception of a master file 400 having a "signed filename" 405 and attached files or links 415 with associated "signed filenames" 420, the method for accessing the documents comprises the steps of :

- analyzing the master file :
  - extracting the encoded digital signature 410 from the "signed filename" 405 of received master file 400, at a predefined position or using delimiters ;
  - recovering the encoded hash value of the received master file 400 by using the public-key of the sender and extracted digital signature 410 ;
  - computing the hash value of received master file 400 by means of a hash function, identical to the hash function used by the sender to compute digital signature 410 ;
  - comparing computed and decoded hash values ; and,
  - if the computed and decoded hash value are identical, processing the received master file 400 as an authentic file else, if the computed and decoded hash value are different, rejecting the received master file 400 as being fake or corrupted ;
- upon selection of an attached or linked file :
  - extracting the encoded digital signature 425 from the "signed filename" 420 of selected attached or linked file 415, at a predefined position or using delimiters ;
  - recovering the encoded hash value of the selected attached or linked file 415 by using the public-key of the sender and extracted digital signature 425 ;
  - computing the hash value of selected attached or linked file 415 by means of a hash function, identical to the hash function used by the sender to compute digital signature 425 ;
  - comparing computed and decoded hash values ; and,
  - if the computed and decoded hash value are identical, processing the selected attached or linked file 415 as an authentic file else, if the computed and decoded

hash value are different, rejecting the selected attached or linked file 415 as being fake or corrupted.

Since the method as described above is adapted for accessing an attached or linked file without analyzing other attached or linked file, it allows to forward independently such attached or linked files, keeping sender authentication information.

Naturally, in order to satisfy local and specific requirements, a person skilled in the art may apply to the solution described above many modifications and alterations all of which, however, are included within the scope of protection of the invention as defined by the following claims.

Claims:

1. A computer file (400) containing digital data characterized in that authentication information (410) is encoded in the filename (405) of said computer file at a predetermined position or using delimiters.

2. A method for encoding authentication information (130) in the filename of a computer file (100) containing digital data, said method comprising the steps of :

- computing a digital signature (130) of the file (100) using a private key (125) of the sender ; and,
- encoding said computed digital signature (130) in the filename of said computer file at a predetermined position or using delimiters.

3. The method of claim 2 wherein said step of encoding said computed digital signature (130) in the filename of said computer file (100) at a predetermined position or using delimiters consists in adding said computed digital signature (130) in the filename just before the file extension, said computed digital signature (130) having a fixed size.

4. The method of either claim 2 or claim 3 wherein said step of computing a digital signature (130) is based on a symmetric or public-key algorithm.

5. The method of anyone of claims 2 to 4 wherein said step of computing a digital signature (130) consists in :

- computing a hash value (115) of said computer file (100) ; and,

- computing a digital signature (130) of said computed hash value (115) using a private key (125) of the sender.

6. The method of claim 5 wherein said step of computing a hash value (115) is based on Secure Hash Algorithm or Message-Digest-5 algorithms (110).

7. A method for authenticating a computer file (200) having a filename (205) comprising authentication information (210) according to either claim 5 or claim 6, said method comprising the steps of :

- extracting said authentication information (210) from said filename (205) of said computer file (200), at a predetermined position or using delimiters ;

- recovering the encoded hash value (220) of the computer file (200) by using the public-key (215) of the sender and said extracted authentication information (210) ;

- computing the hash value (230) of said computer file (200) by means of a hash function (225), identical to the one used for authentication information encoding (110) ;

- comparing said encoded and said computed hash values (220, 230) ; and,

- if said encoded and said computed hash values (220, 230) are identical, processing said computer file (200) else, if said encoded and said computed hash values (220, 230) are different, rejecting said computer file (200).

8. The method of anyone of claims 2 to 7 further comprising the step of applying said method on files attached or linked to said computer file.

9. An apparatus comprising means adapted for carrying out the method according to anyone of the claims 2 to 8.

10. A computer-like readable medium comprising instructions for carrying out the method according to anyone of the claims 2 to 8.

METHOD AND SYSTEMS FOR ENCODING SIGNATURES TO AUTHENTICATE FILES

Abstract

A method and systems for verifying the authenticity and integrity of files transmitted through a computer network is disclosed. According to the invention, authentication information is encoded in the filename of the file. In a preferred embodiment, encoding authentication information consists in computing a hash value (115) of the file (100) with a hash function (110), computing a digital signature (130) of this file hash value using a private-key (125) and encoding the computed digital signature (130) in the filename of the file, at a predetermined position or using delimiters, to create a "signed filename" (135). Upon reception of a file with associated "signed filename" (205), the encoded digital signature (210) is extracted from the "signed filename" (205). Then, the encoded hash value (220) of the file is recovered using a public key (215) and extracted digital signature (210) and compared with the hash value (230) computed on the file. If decoded and computed hash values are identical, the received file is processed as an authentic file else, if decoded and computed hash values are different, the received file is rejected as being fake or corrupted.

Figure 1.

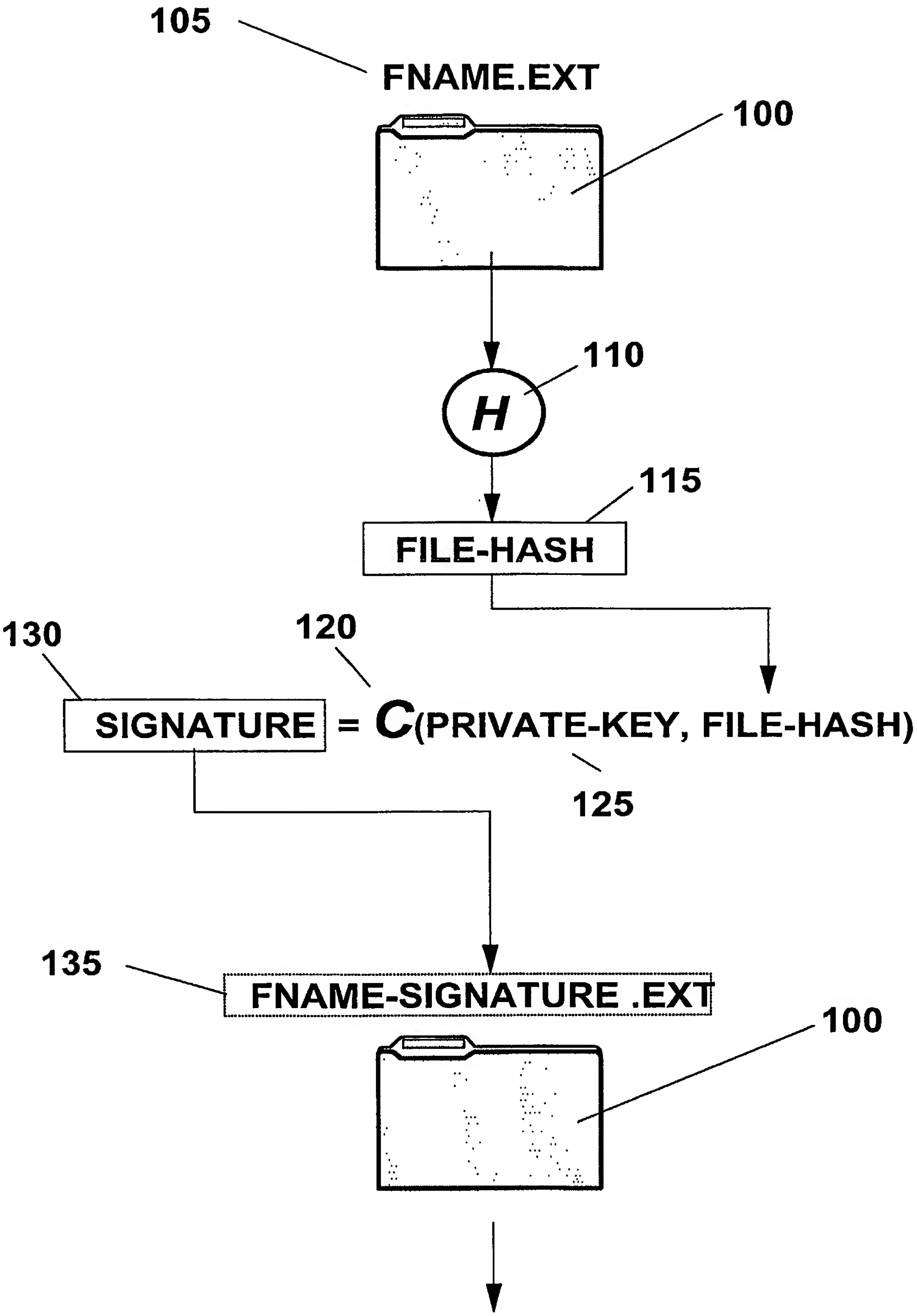

105
FNAME.EXT
100
110
H
115
FILE-HASH
130
120
SIGNATURE = C(PRIVATE-KEY, FILE-HASH)
125
135
FNAME-SIGNATURE .EXT
100

Figure 1

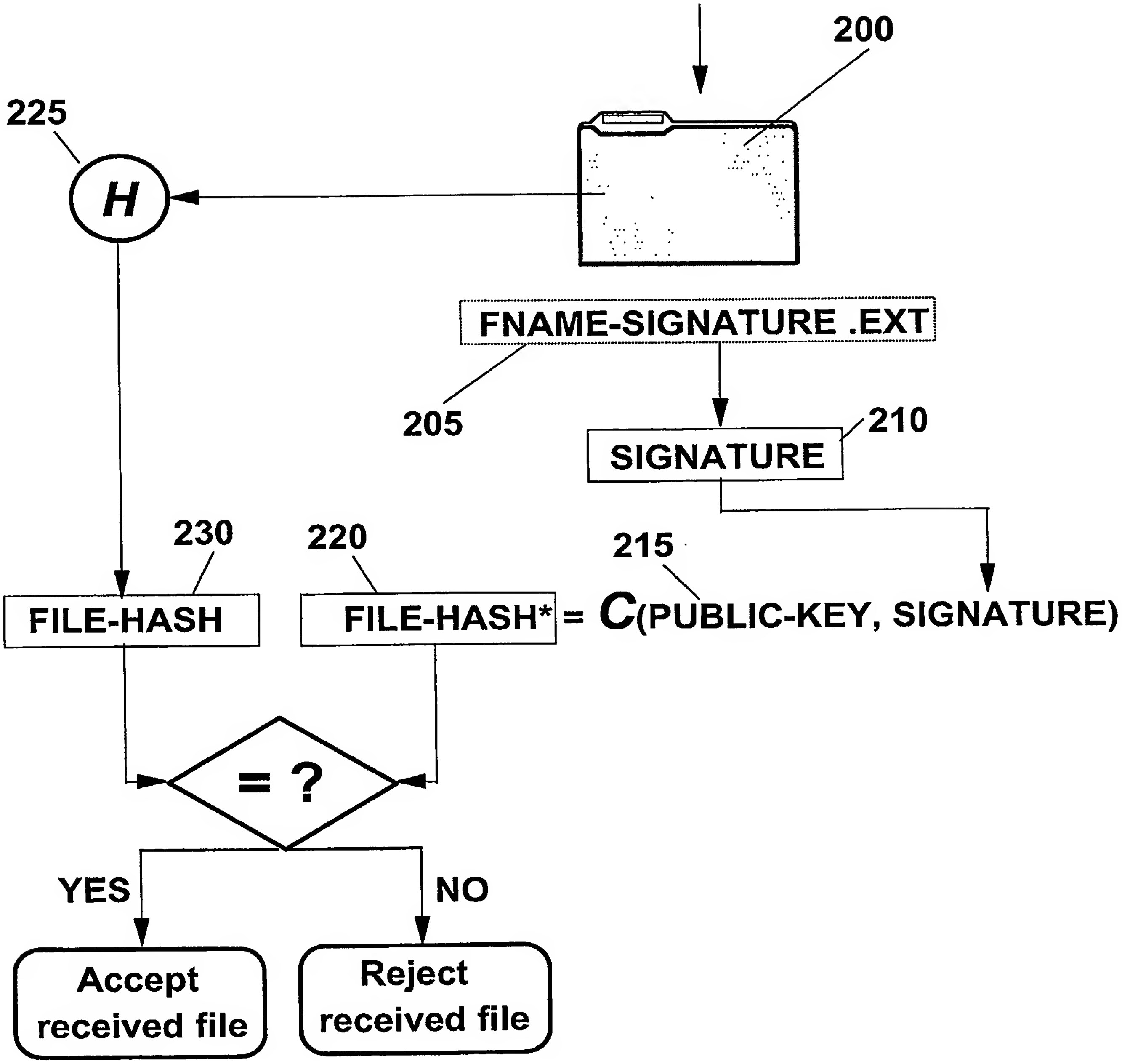
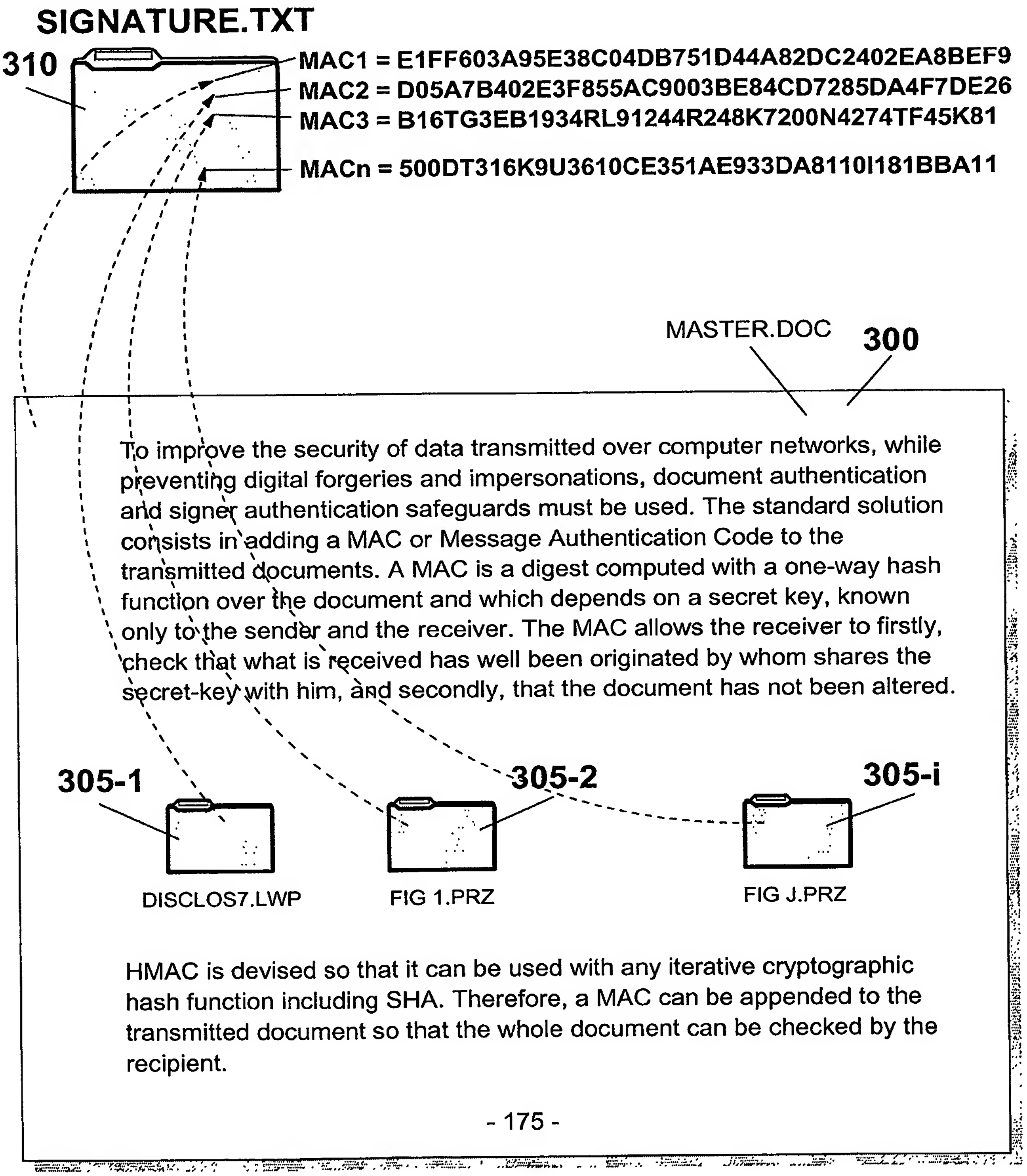

200
225
H
FNAME-SIGNATURE .EXT
205
SIGNATURE
210
230
220
215
FILE-HASH
FILE-HASH* = C(PUBLIC-KEY, SIGNATURE)
= ?
YES
NO
Accept received file
Reject received file

Figure 2


SIGNATURE.TXT
310
MAC1 = E1FF603A95E38C04DB751D44A82DC2402EA8BEF9
MAC2 = D05A7B402E3F855AC9003BE84CD7285DA4F7DE26
MAC3 = B16TG3EB1934RL91244R248K7200N4274TF45K81
MACn = 500DT316K9U3610CE351AE933DA8110I181BBA11
MASTER.DOC
300
305-1
305-2
305-i
DISCLOS7.LWP
FIG 1.PRZ
FIG J.PRZ

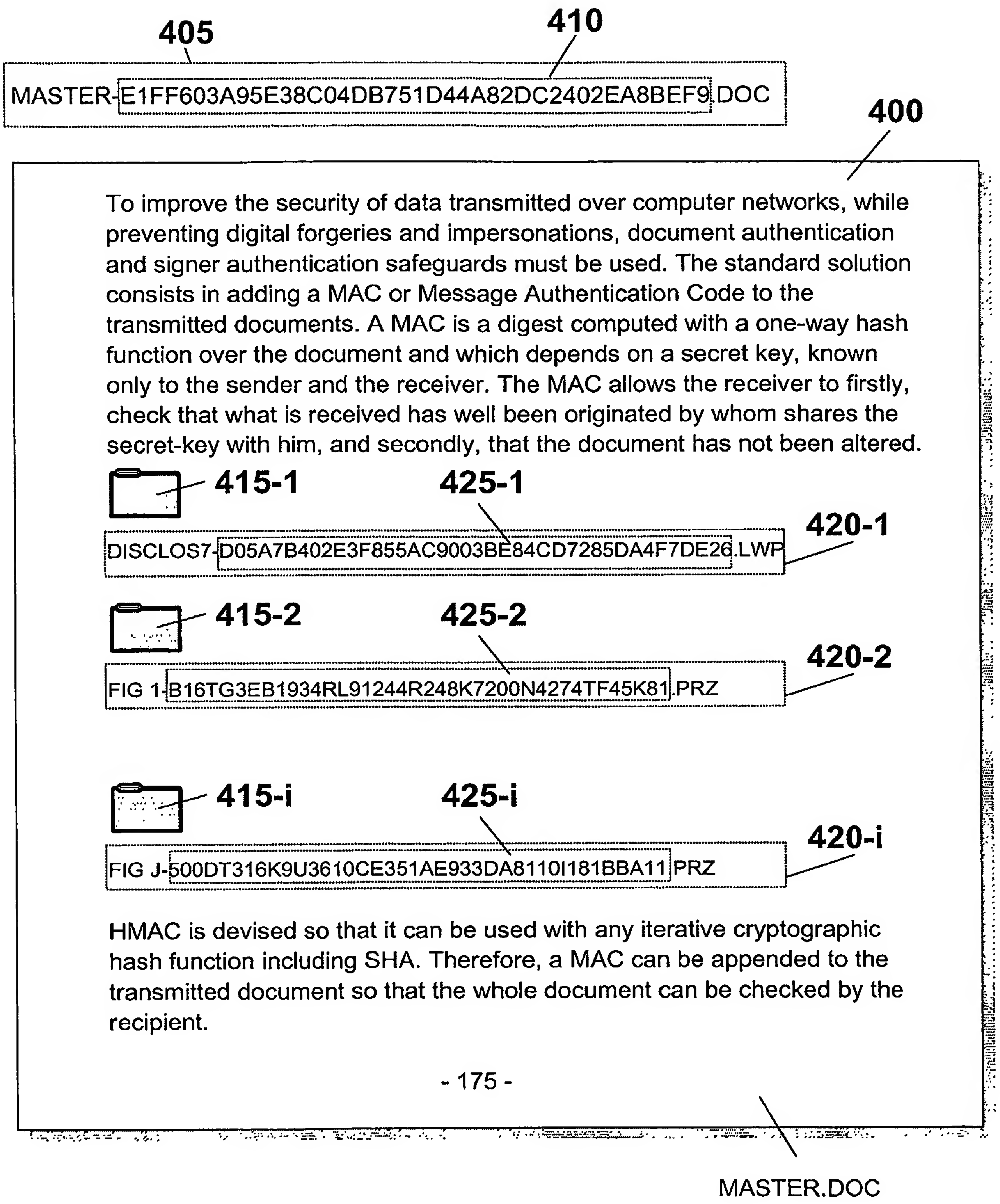
To improve the security of data transmitted over computer networks, while preventing digital forgeries and impersonations, document authentication and signer authentication safeguards must be used. The standard solution consists in adding a MAC or Message Authentication Code to the transmitted documents. A MAC is a digest computed with a one-way hash function over the document and which depends on a secret key, known only to the sender and the receiver. The MAC allows the receiver to firstly, check that what is received has well been originated by whom shares the secret-key with him, and secondly, that the document has not been altered.

HMAC is devised so that it can be used with any iterative cryptographic hash function including SHA. Therefore, a MAC can be appended to the transmitted document so that the whole document can be checked by the recipient.

- 175 -

Figure 3

4/4



MASTER-E1FF603A95E38C04DB751D44A82DC2402EA8BEF9.DOC

To improve the security of data transmitted over computer networks, while preventing digital forgeries and impersonations, document authentication and signer authentication safeguards must be used. The standard solution consists in adding a MAC or Message Authentication Code to the transmitted documents. A MAC is a digest computed with a one-way hash function over the document and which depends on a secret key, known only to the sender and the receiver. The MAC allows the receiver to firstly, check that what is received has well been originated by whom shares the secret-key with him, and secondly, that the document has not been altered.

DISCLOS7-D05A7B402E3F855AC9003BE84CD7285DA4F7DE26.LWP

FIG 1-B16TG3EB1934RL91244R248K7200N4274TF45K81.PRZ

FIG J-500DT316K9U3610CE351AE933DA8110I181BBA11 PRZ

HMAC is devised so that it can be used with any iterative cryptographic hash function including SHA. Therefore, a MAC can be appended to the transmitted document so that the whole document can be checked by the recipient.

- 175 -

Figure 4